

# Distributed Algorithms for Optimal Power Flow Problem

Albert Y.S. Lam, Baosen Zhang, and David Tse

**Abstract**—Optimal power flow (OPF) is an important problem for power generation and it is in general non-convex. With the employment of renewable energy, it will be desirable if OPF can be solved very efficiently so its solution can be used in real time. With some special network structure, e.g. trees, the problem has been shown to have a zero duality gap and the convex dual problem yields the optimal solution. In this paper, we propose a primal and a dual algorithm to coordinate the smaller subproblems decomposed from the convexified OPF. We can arrange the subproblems to be solved sequentially and cumulatively in a central node or solved in parallel in distributed nodes. We test the algorithms on IEEE radial distribution test feeders, some random tree-structured networks, and the IEEE transmission system benchmarks. Simulation results show that the computation time can be improved dramatically with our algorithms over the centralized approach of solving the problem without decomposition, especially in tree-structured problems. The computation time grows linearly with the problem size with the cumulative approach while the distributed one can have size-independent computation time.

## I. INTRODUCTION

A N ELECTRIC power system is the main facility to distribute electricity in modern societies. It is a network connecting power supplies (e.g., thermoelectric generators and turbine steam engines) to consumers. A power grid is generally composed of several subsystems: generation, transmission, substation, distribution, and consumers. The generators generate power which is delivered to the substations at high voltages through the transmission network. The power voltage is stepped down and then distributed to the consumers via the distribution networks. In a typical power system, a few hundreds of generators interconnect to several hundreds of substations. The substations distribute the power to millions of consumers with relatively simpler radial networks with tree-like structures.

In the past, research on power systems mainly focused on the core of the network, i.e., from the generation, via transmission, to the substations. All of the control, planning and optimization was done by a single entity (e.g. an ISO). With the integration of renewal energy and energy storage, self-healing ability, and demand response, the focus is shifted toward the consumer side, i.e. distribution networks, and this new paradigm is called the smart grid [1].

The optimal power flow (OPF) is one of the most important problems in power engineering and it aims to minimize the generation cost subject to demand constraints and the network

physical constraints, e.g. bus voltage limits, bus power limits, thermal line constraint, etc. Due to the quadratic relations between voltage and power, OPF is non-convex. In general, heuristic approaches have been employed to solve the OPF but they are not guaranteed to yield the optimal solution. To simplify the calculation, with assumptions on lossless power line, constant voltage and small voltage angles, OPF can be linearized and this approximation is also called DC-OPF, which is not accurate under all circumstances [2]. For the complex OPF, [3] suggested solving the problem in its dual form and studied the conditions of the power network with zero duality gap. In [4], it was shown that the duality gap is always zero for network structures such as trees which model distribution networks well. [5], as an independent work of this paper, decomposes the OPF in terms of cycles and branches and formulates the problem as an second-order cone program for tree networks which is equivalent to that given in [6]. In traditional power systems, OPF is mainly for planning purpose. For example, it is used to determine the system state in the day-ahead market with the given system information. In the smart grid paradigm, due to highly intermittent nature of the renewable, the later the prediction is made, the more reliable it is. If OPF can be solved very efficiently, we may solve the OPF in real time thus mitigating some of the unpredictability.

We aim at solving OPF efficiently. When the system size (e.g. the number of buses) increases, solving the problem in a centralized manner is not practical (this will be verified in the simulation). One possible way is to tackle the problem distributedly by coordinating several entities in the system, each of which handle part of the problem and their collaborative effort solves the whole problem. To do this, a communication protocol is needed to define what information should be conveyed among the entities. We can learn from the networking protocol development to design a communication protocol for OPF. The earliest form of protocols for the Internet was proposed in 70's. They were designed to handle the increasing volume of traffic sent over the Internet in an ad hoc manner. In 1998, Kelly et al. studied Transmission Control Protocol (TCP), which is one of the core protocols in TCP/IP [7]. They showed that TCP can be analyzed with a fundamental optimization problem for rate control and the algorithms developed from the optimization fit the ad hoc designed variants of TCP. This lays down a framework to design communication protocols for complex systems with reasoning. In this framework, we start with an optimization problem representing the system. By optimization decomposition [8], the problem is decomposed into (sim-

The authors are with the Department of Electrical Engineering and Computer Sciences, University of California, Berkeley, CA, 94720 USA (e-mail: {ayslam, zhangbao, dtse}@eecs.berkeley.edu).

pler) subproblems which can be solved by different entities in the system independently. The coordination between the subproblems define the communication protocols (i.e. what and how the data exchange between the entities). [9] shows that many problems in communications and networking can be cast under this framework and protocols can be designed through primal and dual decomposition. In this paper, we study OPF by decomposing it into subproblems with primal and dual decomposition. Then we propose the primal and dual algorithms, respectively, to solve OPF in a distributed manner and the algorithms determine the communication protocols. Our algorithms do not assume the existence of a communication overlay with topology different from the power network. In other words, a bus only needs to communicate with its one-hop neighbors in the power network. The algorithms can be employed to any power network as long as the strong duality holds. We test the algorithms on IEEE radial distribution test feeders, some random tree-structured networks, and the IEEE transmission system benchmarks. Simulation results show that the algorithms can solve OPF distributively and they are scalable, especially when applied to the distribution network. If we apply the algorithm distributedly, the computational time is independent of the number of buses. If we apply the algorithms (with the problem decomposed) in a central node, the computational time grows linearly with the number of buses.

The rest of this paper is organized as follows. In Section II, we give the OPF formulation and the necessary background. Section III describes the primal and dual algorithms and the mechanism to recover the optimal voltage from the results of the algorithms. We illustrate the algorithms with two examples in Section V and present the simulation results in Section VI. In Section VII, we discuss the characteristics of the algorithms and conclude the paper in Section VIII.

## II. PRELIMINARIES

### A. Problem Formulation

Assume that there are  $n$  buses in the power network. For buses  $i$  and  $k$ ,  $i \sim k$  means that they are connected by a power line and  $i \not\sim k$  otherwise. Let  $z_{ik}$  and  $y_{ik}$  be the complex impedance and admittance between  $i$  and  $k$ , respectively, and we have  $y_{ik} = \frac{1}{z_{ik}}$ . We denote  $\mathbf{Y} = (Y_{ik}, 1 \leq i, k \leq n)$  as the admittance matrix, where

$$Y_{ik} = \begin{cases} \sum_{l \sim i} y_{il} & \text{if } i = k \\ -y_{ik} & \text{if } i \sim k \\ 0 & \text{if } i \not\sim k. \end{cases}$$

Let  $\mathbf{v} = (V_1, V_2, \dots, V_n)^T \in \mathbb{C}^n$  and  $\mathbf{i} = (I_1, I_2, \dots, I_n)^T \in \mathbb{C}^n$  be the voltage and current vectors, respectively. By Ohm's Law and Kirchoff's Current Law, we have  $\mathbf{i} = \mathbf{Y}\mathbf{v}$ . The apparent power injected at bus  $i$  is  $S_i = P_i + jQ_i = V_i I_i^H$ , where  $P_i$  and  $Q_i$  are the real and reactive power, respectively, and  $H$  means Hermitian transpose. We have the real power vector  $\mathbf{p} = (P_1, P_2, \dots, P_n)^T = \text{Re}\{\text{diag}(\mathbf{v}\mathbf{v}^H \mathbf{Y}^H)\}$ , where  $\text{diag}(\mathbf{v}\mathbf{v}^H \mathbf{Y}^H)$  forms a diagonal matrix whose diagonal is  $\mathbf{v}\mathbf{v}^H \mathbf{Y}^H$ . We define the cost function of Bus  $i$  as  $\text{cost}_i(P_i) = c_{i2}P_i^2 + c_{i1}P_i + c_{i0}$ , where  $c_{i0}, c_{i1}, c_{i2} \in \mathbb{R}$  and  $c_{i2} \geq 0, \forall i$ .

OPF can be stated as

$$\underset{i=1}{\text{minimize}} \sum_{i=1}^n \text{cost}_i(P_i) \quad (1a)$$

subject to

$$\underline{V}_i \leq |V_i| \leq \bar{V}_i, \forall i \quad (1b)$$

$$\underline{P}_i \leq P_i \leq \bar{P}_i, \forall i \quad (1c)$$

$$P_{ik} \leq \bar{P}_{ik}, \forall i, k \quad (1d)$$

$$\mathbf{p} = \text{Re}\{\text{diag}(\mathbf{v}\mathbf{v}^H \mathbf{Y}^H)\} \quad (1e)$$

where  $\underline{V}_i, \bar{V}_i, \underline{P}_i, \bar{P}_i$ , and  $\bar{P}_{ik}$  are the lower and upper voltage limits of bus  $i$ , the lower and upper power limits of bus  $i$ , and the real power flow limit between buses  $i$  and  $k$ , respectively. Eq. (1b) is the nodal voltage constraint limiting the magnitude of bus voltage. Eq. (1c) is the nodal power constraint limiting the real power generated or consumed and (1d) is the flow constraint. Eq. (1e) describes the physical properties of the network. In this formulation,  $\mathbf{p}$  and  $\mathbf{v}$  are the variables. Eqs. (1c) and (1d) are box constraints with respect to  $\mathbf{p}$  which are the variables of the objective function (1a) and they are relatively easy to handle. Eq. (1b) together with (1e) make the problem non-convex and hard to solve. To illustrate the algorithms, we first consider a simplified version of OPF with  $c_{i2} = c_{i0} = 0, \forall i$  and neglect (1c) and (1d). Having  $c_{i0} = 0$  will not affect the optimal solution of the original problem. We will explain how to handle non-zero  $c_{i2}$  later. By introducing a  $n \times n$  complex matrix  $W = (W_{ik}, 1 \leq i, k \leq n) = \mathbf{v}\mathbf{v}^H$ , we can write the simplified OPF in the sequel:

$$\underset{i=1}{\text{minimize}} \sum_{i=1}^n c_{i1}P_i \quad (2a)$$

subject to

$$\underline{V}_i^2 \leq W_{ii} \leq \bar{V}_i^2, \forall i \quad (2b)$$

$$\text{rank}(W) = 1 \quad (2c)$$

$$\mathbf{p} = \text{Re}\{\text{diag}(\mathbf{v}\mathbf{v}^H \mathbf{Y}^H)\} \quad (2d)$$

Let  $\mathbf{C} = \text{diag}(c_{11}, c_{21}, \dots, c_{n1})$  and  $\mathbf{M} = (M_{ik}, 1 \leq i, k \leq n) = \frac{1}{2}(\mathbf{Y}^H \mathbf{C} + \mathbf{C} \mathbf{Y})$ . By relaxing the rank constraint (29c), we have the following semidefinite program (SDP):

$$\underset{\mathbf{W}}{\text{minimize}} \text{Tr}(\mathbf{MW}) \quad (3a)$$

subject to

$$\underline{V}_i^2 \leq W_{ii} \leq \bar{V}_i^2, \forall i \quad (3b)$$

$$\mathbf{W} \succeq 0 \quad (3c)$$

where  $\text{Tr}(\cdot)$  is the trace operator. We can solve this SDP at a central control center. However, current algorithms for SDP, e.g. primal-dual interior-point methods [10], can only handle problems with size up to *several hundreds*. We will decompose the problem into smaller ones by exploring the network structure.

### B. Zero Duality Gap

By [4], the simplified OPF and SDP share the equivalent optimal solution provided that the network has a tree structure,

is a lossless cycle, or a combination of tree and cycle. For these kinds of network structures which are typically found in distribution networks, the optimal solution computed from (3) is exactly the same as that from (2). Targeting distribution networks, we can merely focus on (3). For completeness, the approach in [4] is outlined below.

The dual of (3) is given by

$$\begin{aligned} & \text{maximize} \sum_{i=1}^n (-\bar{\lambda}_i \bar{V}_i^2 + \underline{\lambda}_i \underline{V}_i^2) \\ & \text{subject to } \bar{\lambda}_i \geq 0, \underline{\lambda}_i \geq 0 \forall i \\ & \quad \Lambda + M \succcurlyeq 0, \end{aligned} \quad (4)$$

where  $\bar{\lambda}_i$  and  $\underline{\lambda}_i$  are the Lagrangian multipliers associated with the constraints  $W_{ii} \leq \bar{V}_i^2$  and  $\underline{V}_i^2 \leq W_{ii}$  respectively. From the KKT conditions, [4] showed that (3) always has a solution that is rank 1.

### C. Graph Structure

We will use the following graph structures to decompose SDP.

Consider a graph  $G = (V, E)$ , where  $V = \{i | 1 \leq i \leq n\}$  are vertices and  $E = \{(i, k) \in V \times V\}$  are edges. Vertices  $i$  and  $k$  are adjacent if  $(i, k) \in E$ . A clique  $C$  is a subset of  $V$  whose induced subgraph is fully connected, i.e.,  $(i, k) \in E, \forall i, k \in C$ . A clique is maximal if it cannot be extended to form a larger one by including any adjacent vertex to the clique. In other words, there does not exist a clique whose proper subset is a maximal clique. A chord is an edge which connects two non-adjacent vertices in a cycle. A graph is chordal if each of its cycles with four or more vertices contains a chord. Thus a chordal graph does not have a cycle with four or more vertices. If  $G$  is not chordal, we can produce a corresponding chordal graph  $\tilde{G} = (V, \tilde{E})$ , where  $\tilde{E} = E \cup E_f$  and  $E_f = \{(i, j) \in V \times V - E\}$  are chords of  $G$ , called fill-in edges.  $\tilde{G}$  is not unique. From  $\tilde{G}$ , we can compute the set of all possible maximal cliques  $\mathcal{C} = \{C_1, \dots, C_{|\mathcal{C}|}\}$ , where  $C_i = \{j \in V\}$  whose induced subgraph is complete and maximal. If  $G$  is a tree, each pair of vertices connecting by an edge forms a maximal clique. For a tree with  $n$  vertices, it can be decomposed into  $n - 1$  maximal cliques.

For  $\mathbf{M}$  in (3a), we can induce the corresponding  $G$  by having  $V = \{i | 1 \leq i \leq n\}$  and  $E = \{(i, k) | M_{i,k} \neq 0\}$ .  $G$  has a very close relationship with the power network structure because of  $\mathbf{Y}$ . If all  $c_{i1}, \forall i$  are non-zero,  $G$  directly represents the network.

We use the following procedure to produce  $\mathcal{C}$  from  $\mathbf{M}$ .

- 1) Construct a graph  $G = (V, E)$  from  $\mathbf{M}$ .
- 2) From  $G$ , compute Maximum Cardinality Search [11] to construct an elimination ordering  $\sigma$  of vertices [12].
- 3) With  $\sigma$ , perform Fill-In Computation [13] to obtain a chordal graph  $\tilde{G}$ .
- 4) From  $\tilde{G}$ , determine the set of maximal cliques  $\mathcal{C}$  by the Bron-Kerbosch algorithm [14].

Note that similar ideas about maximal cliques have been utilized to develop a parallel IPM for SDP [15]–[17]. However, we make use of the ideas to decompose SDP into smaller problems, which can be tackled by any appropriate SDP

algorithm, not necessarily IPM. Therefore, our approach is more flexible on that any future efficient SDP algorithm can be incorporated into our framework.

## III. ALGORITHMS

In this section, we will first present the primal and dual algorithms whose outputs are positive semidefinite matrices. Then we will explain the mechanism to convert such a matrix into the voltage vector.

### A. Primal Algorithm

The objective function (3a) can be expressed as

$$\text{Tr}(\mathbf{MW}) = \sum_{i,k=1}^n M_{ik}^H W_{ik}, \quad (5)$$

where each term  $M_{ik}^H W_{ik}$  can be classified into one of the following three categories:

- 1) Ignored terms  
Each of which has  $M_{ik} = 0$ . Let  $\mathcal{I} = \{(i, k) | M_{ik} = 0\}$ .
- 2) Unique terms  
For  $M_{ik} \neq 0$ , both  $i$  and  $k$  belong to a unique maximal clique. If  $i, k \in C_l$ , then  $i, k \notin C_r, \forall r \neq l$ . Let  $\mathcal{U} = \{(i, k) | i, k \in C_l, \forall l, i, k \notin C_r, \forall r \neq l\}$ .
- 3) Shared terms  
For  $M_{ik} \neq 0$ , both  $i$  and  $k$  belongs to more than one maximal clique.

Then (5) becomes

$$\begin{aligned} \text{Tr}(\mathbf{MW}) = & \sum_{i,k|(i,k) \in \mathcal{I}} M_{ik}^H W_{ik} + \sum_{i,k|(i,k) \in \mathcal{U}-\mathcal{I}} M_{ik}^H W_{ik} \\ & + \sum_{i,k|(i,k) \notin \mathcal{I} \cup \mathcal{U}} M_{ik}^H W_{ik}, \end{aligned} \quad (6)$$

where all the ignored terms can be ignored. Since each unique term is unique to each maximal clique, (6) becomes

$$\text{Tr}(\mathbf{MW}) = \sum_{\substack{i,k \in C_l, \forall C_l \in \mathcal{C} \\ |(i,k) \in \mathcal{U}-\mathcal{I}}} M_{ik}^H W_{ik} + \sum_{\substack{i,k|(i,k) \notin \mathcal{I} \cup \mathcal{U}}} M_{ik}^H W_{ik}. \quad (7)$$

Eq. (3b) gives bounds to each  $W_{ii}, 1 \leq i \leq n$  and it is equivalent to

$$\bar{V}_i^2 \leq W_{ii} \leq \underline{V}_i^2, \forall i \in C_l, \forall C_l \in \mathcal{C}. \quad (8)$$

By [18], a matrix is positive semidefinite if all its submatrices corresponding to the maximal cliques induced by the matrix are all positive semidefinite. Let  $\mathbf{W}_{C_l C_l}$  be the partial matrix of  $\mathbf{W}$  with rows and columns indexed according to  $C_l$ . Eq. (3c) is equivalent to

$$\mathbf{W}_{C_l C_l} \succeq 0, \forall C_l \in \mathcal{C}. \quad (9)$$

Hence, (3) is written as

$$\text{minimize} \sum_{\substack{i,k \in C_l, \forall C_l \in \mathcal{C} \\ |(i,k) \in \mathcal{U} - \mathcal{I}}} M_{ik}^H W_{ik} + \sum_{i,k|(i,k) \notin \mathcal{I} \cup \mathcal{U}} M_{ik}^H W_{ik} \quad (10a)$$

subject to

$$\underline{V}_i^2 \leq W_{ii} \leq \bar{V}_i^2, \forall i \in C_l, \forall C_l \in \mathcal{C} \quad (10b)$$

$$W_{C_l C_l} \succeq 0, \forall C_l \in \mathcal{C}. \quad (10c)$$

If we fix all  $W_{ik}$  in the shared terms (those in the second summation in (10a)), (10) can be decomposed into  $|\mathcal{C}|$  subproblems, each of which corresponds to a maximal clique. For  $C_l$ , we have the subproblem  $l$ , as follows:

$$\text{minimize} \sum_{i,k \in C_l | (i,k) \in \mathcal{U} - \mathcal{I}} M_{ik}^H W_{ik} \quad (11a)$$

subject to

$$\underline{V}_i^2 \leq W_{ii} \leq \bar{V}_i^2, \forall i \in C_l, i \notin C_r, \forall r \neq l \quad (11b)$$

$$\mathbf{W}_{C_l C_l} \succeq 0. \quad (11c)$$

In (11), only the semidefinite constraint (11c) involves those variables which are not unique to  $C_l$ , i.e.,  $W_{ik}$  such that  $(i,k) \notin \mathcal{U}$ . By introducing a slack variable  $X_{ik,l} = W_{ik}$  for each shared  $W_{ik}$  in subproblem  $l$ , we define  $\tilde{\mathbf{W}}_{C_l C_l} = (\tilde{W}_{ik}, i, k \in C_l)$  and  $\tilde{\mathbf{M}}_{C_l C_l} = (\tilde{M}_{ik}, i, k \in C_l)$  where

$$\tilde{W}_{ik} = \begin{cases} W_{ik} & \text{if } (i,k) \in \mathcal{U} \\ X_{ik,l} & \text{otherwise} \end{cases}$$

and

$$\tilde{M}_{ik} = \begin{cases} M_{ik} & \text{if } (i,k) \in \mathcal{U} \\ 0 & \text{otherwise.} \end{cases}$$

Then (11) becomes

$$\text{minimize} \text{Tr}(\tilde{\mathbf{M}}_{C_l C_l} \tilde{\mathbf{W}}_{C_l C_l}) \quad (12a)$$

subject to

$$\underline{V}_i^2 \leq W_{ii} \leq \bar{V}_i^2, \forall i \in C_l, i \notin C_r, \forall r \neq l \quad (12b)$$

$$\tilde{\mathbf{W}}_{C_l C_l} \succeq 0 \quad (12c)$$

$$X_{ik,l} = W_{ik}, \forall i, k | (i,k) \notin \mathcal{U}. \quad (12d)$$

Note that  $W_{ik}$ 's in (12d) are given to the subproblem. When given such  $W_{ik}$ 's, all subproblems are independent and can be solved in parallel. Let the domain of (12) be  $\Phi_l$ . Given  $W_{ik}$  where  $(i,k) \notin \mathcal{U}$ , let  $\phi_l(W_{ik} | (i,k) \notin \mathcal{U}) = \inf_{\tilde{\mathbf{W}}_{C_l C_l} \in \Phi_k} \{\text{Tr}(\tilde{\mathbf{M}}_{C_l C_l} \tilde{\mathbf{W}}_{C_l C_l})\}$ . (10) becomes

$$\text{minimize} \sum_{\forall C_l \in \mathcal{C}} \phi_l(W_{ik} | (i,k) \notin \mathcal{U}) + \sum_{i,k|(i,k) \notin \mathcal{U}} M_{ik}^H W_{ik} \quad (13a)$$

subject to

$$\underline{V}_i^2 \leq W_{ii} \leq \bar{V}_i^2, \forall i | (i,i) \notin \mathcal{U}. \quad (13b)$$

Eq. (13) is the master problem which minimizes those  $W_{ik}$  shared by the maximal cliques. With those shared  $W_{ik}$  computed in (13), we minimize the  $W_{ik}$  unique to each subproblem given in (12).

In (13), those shared  $W_{ik}$  can be further classified according to nodes and edges:

1) Nodes

Let  $\lambda_{ii,l}$  be the Lagrangian multiplier for (12d) with  $i = k$ . The subgradient of  $W_{ii}$  with respect to subproblem  $l$  is  $-\lambda_{ii,l}$  [19]. Thus the overall subgradient is  $\sum_{l|i \in C_l} (-\lambda_{ii,l}) + M_{ii}^H$ . At iteration  $t$ , we update  $W_{ii}$  by

$$W_{ii}^{(t+1)} = \text{Proj} \left( W_{ii}^{(t)} - \alpha^{(t)} \left( \sum_{l|i \in C_l} (-\lambda_{ii,l}) + M_{ii}^H \right) \right), \quad (14)$$

where

$$\text{Proj}(x) = \begin{cases} \frac{\underline{V}_i^2}{\bar{V}_i^2} & \text{if } x < \frac{\underline{V}_i^2}{\bar{V}_i^2}, \\ \frac{\bar{V}_i^2}{\underline{V}_i^2} & \text{if } x > \frac{\bar{V}_i^2}{\underline{V}_i^2}, \\ x & \text{otherwise,} \end{cases}$$

$\alpha^{(t)}$  is the step size at iteration  $t$ , and  $W_{ii}^{(t)}$  represents  $W_{ii}$  at iteration  $t$ .

2) Edges in  $E$

We consider (12d) with  $i \neq k$ . Since  $W_{ik}$  and  $X_{ik,l}$  are complex numbers, we can handle the real and imaginary parts separately, i.e.  $\text{Re}\{X_{ik,l}\} = \text{Re}\{W_{ik}\}$  and  $\text{Im}\{X_{ik,l}\} = \text{Im}\{W_{ik}\}$ . Let  $\lambda_{ik,l}^{\text{Re}}$  and  $\lambda_{ik,l}^{\text{Im}}$  be their corresponding Lagrangian multipliers for subproblem  $l$ , respectively. In (20a),  $\forall i \neq k$ , the  $ik$  and  $ki$  terms always come in a pair. We have

$$M_{ik}^H W_{ik} + M_{ki}^H W_{ki} = 2\text{Re}\{M_{ik}^H\} \text{Re}\{W_{ik}\} - 2\text{Im}\{M_{ik}^H\} \text{Im}\{W_{ik}\}.$$

A subgradient of the real part of  $W_{ik}$  is  $\sum_{l|i,k \in C_l} (-\lambda_{ik,l}^{\text{Re}}) + 2\text{Re}\{M_{ik}^H\}$ . At iteration  $t$ , we update  $\text{Re}\{W_{ik}^{(t)}\}$  by

$$\text{Re}\{W_{ik}^{(t+1)}\} = \text{Re}\{W_{ik}^{(t)}\} - \alpha^{(t)} \left( \sum_{l|i,k \in C_l} (-\lambda_{ik,l}^{\text{Re}}) + 2\text{Re}\{M_{ik}^H\} \right). \quad (15)$$

Similarly, for the imaginary part, we have

$$\text{Im}\{W_{ik}^{(t+1)}\} = \text{Im}\{W_{ik}^{(t)}\} - \alpha^{(t)} \left( \sum_{l|i,k \in C_l} (-\lambda_{ik,l}^{\text{Im}}) - 2\text{Im}\{M_{ik}^H\} \right). \quad (16)$$

3) Edges in  $E_f$

Recall that fill-in edges are “artificial” edges added to  $G$  to make  $\tilde{G}$ . For  $(i,k) \in E_f$ , we have  $M_{ik} = 0, i \neq k$ . Similarly, at iteration  $k$ , we update its real and imaginary parts by

$$\text{Re}\{W_{ik}^{(t+1)}\} = \text{Re}\{W_{ik}^{(t)}\} - \alpha^{(t)} \left( \sum_{l|i,k \in C_l} (-\lambda_{ik,l}^{\text{Re}}) \right), \quad (17)$$

and

$$\text{Im}\{W_{ik}^{(t+1)}\} = \text{Im}\{W_{ik}^{(t)}\} - \alpha^{(t)} \left( \sum_{l|i,k \in C_l} (-\lambda_{ik,l}^{\text{Im}}) \right). \quad (18)$$

We can interpret the updating mechanism as follows: certain maximal cliques share a component  $W_{ik}$  (if  $i = k$ , it corresponds to a node; otherwise, it corresponds to an edge or a fill-in edge).  $W_{ik}$  represents electricity resources and  $-M_{ik}^H$  is its default price. An agent (i.e. a node responsible for computing the update) which is common to all those maximal cliques sharing the resource determines how much resource should be allocated to each maximal clique. In other words, it fixes  $W_{ik}$  and every party gets this amount.  $X_{ik,l}$  is the actual resource required by  $C_l$  and  $\lambda_{ik,l}$  corresponds to the price of the resources when  $W_{ik}$  is allocated to it. If  $C_l$  requires more resource than those allocated, i.e.,  $X_{ik,l} > W_{ik}$ , then  $\lambda_{ik,l} > 0$ . If the net price, i.e.  $\sum_{l|i,k \in C_l} \lambda_{ik,l} - M_{ik}$ , is positive, the agent should increase the amount of resource allocating to the maximal cliques because it can earn more. If the net price is negative, then supply is larger than demand and it should reduce the amount of allocated resources.

From (14)–(18), all shared  $W_{ik}$  can be updated independently. The update of each  $W_{ik}$  only involves those  $\{C_l | C_l \in \mathcal{C}, i, k \in C_l\}$ . In other words, (13) can be further computed separately according to those maximal cliques shared by each  $W_{ik}$ .

The pseudocode of the primal algorithm is as follows:

---

#### Algorithm 1 Primal Algorithm

---

Given  $Q, \bar{V}, \underline{V}, \mathcal{C}$

1. Construct (12) for each maximal clique
  2. **while** stopping criteria not matched **do**
  3. **for** each subproblem  $l$  (in parallel) **do**
  4. Given  $W_{ik}$  with  $(i, k) \notin \mathcal{U}$ , solve (12)
  5. Return  $\lambda_{ik,l} \forall i, k | (i, k) \notin \mathcal{U}$
  6. **end for**
  7. Given  $\lambda_{ik,l} \forall l | i, k \in C_l$ , update the shared  $W_{ik}$  with (14)–(18) (in parallel)
  8. **end while**
- 

#### B. Dual Algorithm

Let  $\Omega_{ik} = \{C_l | i, k \in C_l, \forall l\}$ . Problem (5) can be written as

$\text{Tr}(\mathbf{MW})$

$$\begin{aligned} &= \sum_{i,k|(i,k) \in \mathcal{U}} M_{ik}^H W_{ik} + \sum_{i,k|(i,k) \notin \mathcal{U}} |\Omega_{ik}| \frac{M_{ik}^H W_{ik}}{|\Omega_{ik}|} \\ &= \sum_{C_l \in \mathcal{C}} \left( \sum_{i,k \in C_l | (i,k) \in \mathcal{U}} M_{ik}^H W_{ik} + \sum_{i,k \in C_l | (i,k) \notin \mathcal{U}} \frac{M_{ik}^H W_{ik}}{|\Omega_{ik}|} \right). \end{aligned} \quad (19)$$

Problem (3) becomes

$$\text{minimize} \sum_{C_l \in \mathcal{C}} \left( \sum_{\substack{i,k \in C_l \\ |(i,k) \in \mathcal{U}}} M_{ik}^H W_{ik} + \sum_{\substack{i,k \in C_l \\ |(i,k) \notin \mathcal{U}}} \frac{M_{ik}^H W_{ik}}{|\Omega_{ik}|} \right) \quad (20a)$$

subject to

$$\underline{V}_i^2 \leq W_{ii} \leq \bar{V}_i^2, \forall i \in C_l, \forall C_l \in \mathcal{C} \quad (20b)$$

$$\mathbf{W}_{C_l C_l} \succeq 0, \forall C_l \in \mathcal{C}. \quad (20c)$$

Problem (20) can be separated into subproblems based on the maximal cliques. However, the subproblems are not completely independent of each other as (20c) involves some common variables shared between the subproblems. Similar to the primal algorithm, we can replace  $\mathbf{W}_{C_l C_l}$  with  $\tilde{\mathbf{W}}_{C_l C_l}$ . For each  $W_{ik}|(i, k) \notin \mathcal{U}$ , let  $X_{ik,l}$  be a copy of  $W_{ik}$  in  $C_l \in \Omega_{ik}$ . To make all  $\mathbf{W}_{C_l C_l}$  consistent, we should have

$$\begin{aligned} W_{ik} = X_{ik,l_1} = X_{ik,l_2} = \cdots = X_{ik,l_{|\Omega_{ik}|}}, \forall l_r | C_{l_r} \in \Omega_{ik}, \\ \forall W_{ik}|(i, k) \notin \mathcal{U}, \end{aligned} \quad (21)$$

or simply

$$\begin{aligned} X_{ik,l_1} = X_{ik,l_2} = \cdots = X_{ik,l_{|\Omega_{ik}|}}, \forall l_r | C_{l_r} \in \Omega_{ik}, \\ \forall i, k | (i, k) \notin \mathcal{U}. \end{aligned} \quad (22)$$

For each  $(i, k) \notin \mathcal{U}$ , (22) can be written into  $|\Omega_{ik}| - 1$  equalities, e.g.,

$$\begin{aligned} X_{ik,l_1} &= X_{ik,l_2}, \\ X_{ik,l_2} &= X_{ik,l_3}, \\ &\vdots \\ X_{ik,l_{|\Omega_{ik}|-1}} &= X_{ik,l_{|\Omega_{ik}|}}. \end{aligned} \quad (23)$$

As shown later, the update mechanism of the dual algorithm depends only on how we arrange (22) into equalities. In fact, there are many ways to express the  $|\Omega_{ik}| - 1$  equalities provided that each  $X_{ik,l}$  appears in at least one of the equalities. Suppose the  $r$ th equality be  $\tilde{X}_{ik,r}(1) = \tilde{X}_{ik,r}(2)$ . We assign a Lagrangian multiplier  $v_{ik,r}$  to it. Then we have

$$\begin{aligned} v_{ik,1}(\tilde{X}_{ik,1}(1) - \tilde{X}_{ik,1}(2)) &= 0 \\ v_{ik,2}(\tilde{X}_{ik,2}(1) - \tilde{X}_{ik,2}(2)) &= 0 \\ &\vdots \\ v_{ik,|\Omega_{ik}|-1}(\tilde{X}_{ik,|\Omega_{ik}|-1}(1) - \tilde{X}_{ik,|\Omega_{ik}|-1}(2)) &= 0 \end{aligned} \quad (24)$$

When we sum all these equalities up, each  $X_{ik,l}$  will be associated with an aggregate Lagrangian multiplier  $\tilde{v}_{ik,l}$ , which is composed of all  $v_{ik,r}$  associated with  $X_{ik,l}$ . For example, in (23), we have  $\tilde{X}_{ik,1}(1) = X_{ik,l_1}$ ,  $\tilde{X}_{ik,1}(2) = X_{ik,l_2}$ , and  $\tilde{X}_{ik,2}(1) = X_{ik,l_2}$ . Thus  $\tilde{v}_{ik,l_1} = v_{ik,1}$  and  $\tilde{v}_{ik,l_2} = v_{ik,2} - v_{ik,1}$ .

In (24),  $\forall 1 \leq i, k \leq n$ , the corresponding  $r$ th equality for the  $(i, k)$  pair implies the one for the  $(k, i)$  pair, i.e.,

$$\tilde{X}_{ik,r}(1) = \tilde{X}_{ik,r}(2) \Rightarrow \tilde{X}_{ki,r}(1) = \tilde{X}_{ki,r}(2),$$

due the positive semidefinite property of  $\mathbf{W}$  given in (3c). We have

$$\begin{aligned} v_{ik,r}\tilde{X}_{ik,r}(1) &= v_{ik,r}\tilde{X}_{ik,r}(2) \\ \Rightarrow (v_{ik,r}\tilde{X}_{ik,r}(1))^H &= (v_{ik,r}\tilde{X}_{ik,r}(2))^H \\ \Rightarrow v_{ik,r}^H\tilde{X}_{ik,r}^H(1) &= v_{ik,r}^H\tilde{X}_{ik,r}^H(2) \\ \Rightarrow v_{ik,r}^H\tilde{X}_{ki,r}(1) &= v_{ik,r}^H\tilde{X}_{ki,r}(2). \end{aligned}$$

Thus the aggregate Lagrangian multiplier for  $X_{ki,l}$  can be computed directly from that for  $X_{ik,l}$ , i.e.,  $\tilde{v}_{ki,l} = \tilde{v}_{ik,l}^H$ .

Let  $\tilde{v} = (\tilde{v}_{ik,l}, i, k \in C_l | (i, k) \notin \mathcal{U}, \forall C_l \in \mathcal{C}; l_r | C_{l_r} \in \Omega_{ik})$ . We form the dual function  $d(\tilde{v}, \mathbf{W})$  by aggregating (22) into (19). We have

$$\begin{aligned} d(\tilde{v}, \mathbf{W}) &= \sum_{C_l \in \mathcal{C}} \left( \sum_{i, k \in C_l | (i, k) \in \mathcal{U}} M_{ik}^H W_{ik} + \sum_{i, k \in C_l | (i, k) \notin \mathcal{U}} \frac{M_{ik}^H W_{ik}}{|\Omega_{ik}|} \right) \\ &\quad + \sum_{i, k | (i, k) \notin \mathcal{U}} \sum_{r=1 | C_{l_r} \in \Omega_{ik}}^{| \Omega_{ik} |} \tilde{v}_{ik,l_r} X_{ik,l_r} \\ &= \sum_{C_l \in \mathcal{C}} \left( \sum_{\substack{i, k \in C_l \\ |(i, k) \in \mathcal{U}}} M_{ik}^H W_{ik} + \sum_{\substack{i, k \in C_l \\ |(i, k) \notin \mathcal{U}}} \left( \frac{M_{ik}^H W_{ik}}{|\Omega_{ik}|} + \tilde{v}_{ik,l} X_{ik,l} \right) \right) \\ &\triangleq \sum_{C_l \in \mathcal{C}} d(\tilde{v}, \tilde{\mathbf{W}}_{C_l C_l}) \end{aligned} \quad (25)$$

Given  $\tilde{v}$ , (20) becomes

$$\text{minimize}_{C_l \in \mathcal{C}} \sum_{C_l \in \mathcal{C}} d(\tilde{v}, \tilde{\mathbf{W}}_{C_l C_l}) \quad (26a)$$

subject to

$$\underline{V}_i^2 \leq W_{ii} \leq \bar{V}_i^2, \forall i \in C_l, \forall C_l \in \mathcal{C} \quad (26b)$$

$$\tilde{\mathbf{W}}_{C_l C_l} \succeq 0, \forall C_l \in \mathcal{C}. \quad (26c)$$

Then problem (26) can be divided into subproblems according to the maximal cliques and each of them is independent of each other. Subproblem  $l$  is stated as:

$$\text{minimize}_{\substack{i, k \in C_l \\ |(i, k) \in \mathcal{U}}} M_{ik}^H W_{ik} + \sum_{\substack{i, k \in C_l \\ |(i, k) \notin \mathcal{U}}} \left( \frac{M_{ik}^H}{|\Omega_{ik}|} + \tilde{v}_{ik,l} \right) X_{ik,l} \quad (27a)$$

subject to

$$\underline{V}_i^2 \leq W_{ii} \leq \bar{V}_i^2, \forall i \in C_l | (i, i) \in \mathcal{U}, \quad (27b)$$

$$\underline{V}_i^2 \leq X_{ii,l} \leq \bar{V}_i^2, \forall i \in C_l | (i, i) \notin \mathcal{U}, \quad (27c)$$

$$\tilde{\mathbf{W}}_{C_l C_l} \succeq 0. \quad (27d)$$

By solving (27), we denote the optimal  $\tilde{X}_{ik,r}(z)$  for the  $r$ th equality in (24) by  $\tilde{X}_{ik,r}^{opt}(z)$ , where  $z \in \{1, 2\}$ . The gradient of  $-d(\tilde{v}, \tilde{\mathbf{W}}_{C_l C_l})^1$  with respect to  $v_{ik,r}$  is

$$-\frac{\partial d}{\partial v_{ik,r}} = \tilde{X}_{ik,r}^{opt}(2) - \tilde{X}_{ik,r}^{opt}(1).$$

<sup>1</sup>In the dual form, we maximize  $\inf_{\tilde{\mathbf{W}}} d(\tilde{v}, \tilde{\mathbf{W}})$  over  $\tilde{v}$ . In minimization, we consider  $-d(\tilde{v}, \tilde{\mathbf{W}}_{C_l C_l})$ .

Let  $v_{ik,r}^{(t)}, \alpha^{(t)} > 0$ , and  $\tilde{X}_{ik,r}^{(t)}(z)$  be the Lagrangian multiplier of the  $r$ th equality associated with  $W_{ik}$ , the step size,  $\tilde{X}_{ik,r}^{opt}(z)$ , respectively, at time  $t$ . Then we can update  $v_{ik,r}$  in (24) by

$$v_{ik,r}^{(t+1)} = v_{ik,r}^{(t)} - \alpha^{(t)} \left( \tilde{X}_{ik,r}^{(t)}(2) - \tilde{X}_{ik,r}^{(t)}(1) \right). \quad (28)$$

If  $\tilde{X}_{ik,r}^{(t)}(2) < \tilde{X}_{ik,r}^{(t)}(1)$ , then  $v_{ik,r}^{(t+1)} > v_{ik,r}^{(t)}$ . This will make the coefficient corresponding to  $\tilde{X}_{ik,r}^{(t)}(2)$  larger while making that corresponding to  $\tilde{X}_{ik,r}^{(t)}(1)$  smaller. At time  $t+1$ , the subproblem will obtain  $\tilde{X}_{ik,r}^{(t+1)}(1) < \tilde{X}_{ik,r}^{(t)}(1)$  and  $\tilde{X}_{ik,r}^{(t+1)}(2) > \tilde{X}_{ik,r}^{(t)}(2)$ . Hence,  $|\tilde{X}_{ik,r}^{(t+1)}(2) - \tilde{X}_{ik,r}^{(t+1)}(1)| < |\tilde{X}_{ik,r}^{(t)}(2) - \tilde{X}_{ik,r}^{(t)}(1)|$ . On the other hand, if  $\tilde{X}_{ik,r}^{(t)}(2) > \tilde{X}_{ik,r}^{(t)}(1)$ , then  $v_{ik,r}^{(t+1)} < v_{ik,r}^{(t)}$ . This will make the coefficient corresponding to  $\tilde{X}_{ik,r}^{(t)}(1)$  smaller while making that corresponding to  $\tilde{X}_{ik,r}^{(t)}(2)$  larger. Then we will get  $\tilde{X}_{ik,r}^{(t+1)}(1) > \tilde{X}_{ik,r}^{(t)}(1)$  and  $\tilde{X}_{ik,r}^{(t+1)}(2) < \tilde{X}_{ik,r}^{(t)}(2)$ . This will also make  $|\tilde{X}_{ik,r}^{(t+1)}(2) - \tilde{X}_{ik,r}^{(t+1)}(1)| < |\tilde{X}_{ik,r}^{(t)}(2) - \tilde{X}_{ik,r}^{(t)}(1)|$ . Therefore, (28) drives  $X_{ik,l}$ 's in (22) become closer to each other in value when the algorithm evolves. In other words, (28) tries to make equality (22) hold when the algorithm converges.

At any time before the algorithm converges, i.e., (22) does not hold, the solution  $\mathbf{W}$  with the computed  $X_{ik,l}, \forall i, k | (i, k) \notin \mathcal{U}, \forall l | C_l \in \Omega_{ik}$  is an infeasible solution. We can always construct a feasible  $\hat{\mathbf{W}}$  with  $W_{ik}$  which is the average of all  $X_{ik,l}$  in (22).

The purpose of (28) is to make the two entity  $\tilde{X}_{ik,r}^{(t)}(1)$  and  $\tilde{X}_{ik,r}^{(t)}(2)$  closer to each other. As long as  $\tilde{X}_{ik,r}^{(t)}(1)$  and  $\tilde{X}_{ik,r}^{(t)}(2)$  have been computed (from two subproblems), we can update  $v_{ik,r}$  with 28. Thus different  $v_{ik,r}$  can be updated asynchronously. Since the only co-ordination between subproblems is through (28), synchronization is not required in dual algorithm.

We can interpret the updating mechanism as follows:  $v_{ik,r}$  is the price assigned to equality  $\tilde{X}_{ik,r}(1) = \tilde{X}_{ik,r}(2)$ . We can treat  $\tilde{X}_{ik,r}(1)$  and  $\tilde{X}_{ik,r}(2)$  as demand and supply of electricity resources, respectively. If the demand is larger than the supply, i.e.,  $\tilde{X}_{ik,r}(1) > \tilde{X}_{ik,r}(2)$ , we should increase the price so as to suppress the demand and to equalize the supply and demand. On the other hand, if the supply is larger than the demand, i.e.  $\tilde{X}_{ik,r}(1) < \tilde{X}_{ik,r}(2)$ , we should reduce the price in order to boost the demand.

The pseudocode of the dual algorithm is as follows:

### C. Computation of Voltage

When either the primal or the dual algorithm converges, assuming zero duality gap, we obtain the optimal  $\mathbf{W} = \mathbf{v}\mathbf{v}^H$ . To obtain each bus voltage and voltage flown on each line, we first compute the voltage magnitude at each bus,  $|V_i| = \sqrt{W_{ii}}, 1 \leq i \leq n$ . For  $1 \leq i, k \leq n$ , if there is a line between nodes  $i$  and  $k$ , the corresponding line voltage angle difference  $\theta_{ik}$  can be found by solving  $W_{ik} = |V_i||V_k|e^{j\theta_{ik}}$  at either bus  $i$  or  $k$ . For the former, bus  $k$  needs to send  $|V_k|$  to bus  $i$ , and vice versa. By fixing the voltage angle of a particular bus to zero, the voltages of the whole network can be found subsequently.

---

**Algorithm 2** Dual Algorithm

---

Given  $Q, \bar{V}, \underline{V}, \mathcal{C}$

1. Pair up slack variables for the shared variables into equalities
  2. Construct (27) for each maximal clique
  3. **while** stopping criteria not matched **do**
  4. **for** each subproblem  $l$  (in parallel) **do**
  5. Given  $\tilde{v}_{ik}$ , solve (27)
  6. Return  $X_{ik,l}, \forall i, k | (i, k) \notin \mathcal{U}$
  7. **end for**
  8. Given  $X_{ik,l_r}$ , update the price  $v_{ik,l_r}$  with (28) (in parallel and asynchronously)
  9. **end while**
- 

#### IV. QUADRATIC COST FUNCTION

Up to now we have focused on the OPF problem with a linear objective function. In practice, sometimes a quadratic cost function is used. If this is the case, the methods developed so far can be used as subroutines to solve the OPF problem by adding a outer loop to the iteration.

Let  $\text{cost}_i(P_i) = c_{i2}P_i^2 + c_{i1}P_i$  be the cost function associated with  $P_i$ . We assume this function is convex for all buses, that is,  $c_{i2} > 0 \forall i$ . From (1e),  $P_i = \text{Tr}(\mathbf{A}_i \mathbf{v} \mathbf{v}^H)$ , where  $\mathbf{A}_i = \frac{1}{2}((Y^H E_i) + E_i Y)$  and  $E_i$  is the matrix with 1 in the  $(i, i)$ th entry and zero everywhere else. Now the OPF problem is (compare with (2))

$$\underset{i=1}{\text{minimize}} \sum_{i=1}^n c_{i2} \text{Tr}(A_i W)^2 + c_{i1} \text{Tr}(A_i W) \quad (29a)$$

subject to

$$\underline{V}_i^2 \leq W_{ii} \leq \bar{V}_i^2, \forall i \quad (29b)$$

$$\text{rank}(W) = 1 \quad (29c)$$

$$\mathbf{p} = \text{Re}\{\text{diag}(\mathbf{v} \mathbf{v}^H \mathbf{Y}^H)\} \quad (29d)$$

. Using Schur's complement, we may write (29) equivalently as

$$\underset{i=1}{\text{minimize}} \sum_{i=1}^n (t_i + c_{i1} \text{Tr}(A_i W)) \quad (30a)$$

$$\text{subject to } \begin{bmatrix} t_i & \sqrt{c_{i2}} \text{Tr}(\mathbf{A}_i \mathbf{W}) \\ \sqrt{c_{i2}} \text{Tr}(\mathbf{A}_i \mathbf{W}) & 1 \end{bmatrix} \succeq 0 \forall i \quad (30b)$$

$$\underline{V}_i^2 \leq W_{ii} \leq \bar{V}_i^2, \forall i$$

$$\text{rank}(W) = 1.$$

Relax the the rank 1 constraint and taking the dual, we get

$$\underset{i=1}{\text{maximize}} \sum_{i=1}^n (-\bar{\lambda}_i \bar{V}_i^2 + \underline{\lambda}_i \underline{V}_i^2 - u_i) \quad (31a)$$

$$\text{subject to } \sum_{i=1}^n (c_{i1} A_i - 2\sqrt{c_{i2}} z_i A_i) + \Lambda \succcurlyeq 0 \quad (31b)$$

$$\begin{bmatrix} 1 & z_i \\ z_i & u_i \end{bmatrix} \succcurlyeq 0 \forall i, \quad (31c)$$

where the constraint (31c) corresponds to the Schur's complement constraint in (30). The constraint (31c) can be rewritten as  $u_i \geq z_i^2$ , for a given  $z_i$ , the maximizing  $u_i$  is  $z_i^2$ . Therefore

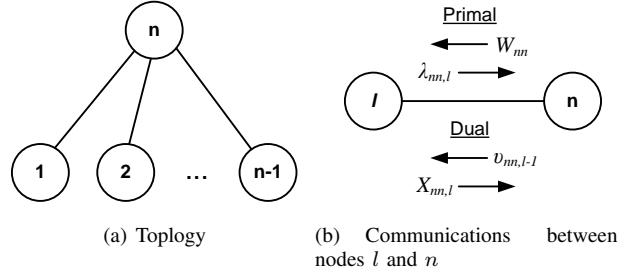


Fig. 1.  $n$ -bus radial network

the we may drop the constraints (31c) and replace the  $u_i$  in the objective function by  $z_i^2$ . If we fix the  $z_i$ 's, then (31) becomes a function of  $z_i$ 's

$$\begin{aligned} J(\mathbf{z}) = & \underset{i=1}{\text{maximize}} \sum_{i=1}^n (-\bar{\lambda}_i \bar{V}_i^2 + \underline{\lambda}_i \underline{V}_i^2 - z_i^2) \\ \text{subject to } & \sum_{i=1}^n (c_{i1} A_i - 2\sqrt{c_{i2}} z_i A_i) + \Lambda \succcurlyeq 0. \end{aligned} \quad (32)$$

For fixed  $\mathbf{z}$ , (32) is in the form of (4). Therefor  $J(\mathbf{z})$  is a dual of the optimization problem with linear cost functions with costs  $(c_{11} - 2\sqrt{c_{12}} z_1, c_{21} - 2\sqrt{c_{22}} z_2, \dots, c_{n1} - 2\sqrt{c_{n2}} z_n)$ . To find the optimal solution of (32) we may use any of the algorithm in the previous sections. Let  $W^*(\mathbf{z})$  denote the optimal solution to  $J(\mathbf{z})$ . To find the optimal  $\mathbf{z}$ , we use a gradient algorithm. The Lagrangian of (32) is

$$\begin{aligned} \mathcal{L}(\lambda, W) = & \sum_{i=1}^n (-\bar{\lambda}_i \bar{V}_i^2 + \underline{\lambda}_i \underline{V}_i^2 - z_i^2) \\ & + \text{Tr}((\sum_{i=1}^n (c_{i1} A_i - 2\sqrt{c_{i2}} z_i A_i) + \Lambda) W). \end{aligned} \quad (33)$$

By a standard result in convex programming, the gradient of  $J(z)$  is given by  $\frac{J(\mathbf{z})}{\partial z_i} = \frac{\partial \mathcal{L}(\lambda^*, W^*)}{\partial z_i} = -2\text{Tr}(\sqrt{c_{i2}} A_i W^*) - 2z_i$ , where  $(\lambda^*, W^*)$  is a pair of optimal dual-primal solutions (dependent on  $\mathbf{z}$ ). Therefore, to solve the problem with quadratic cost functions, we add an additional outer loop to the solution algorithms for the linear cost functions.

#### V. ILLUSTRATIVE EXAMPLES

In this section, we will consider two examples to illustrate how the algorithms work. The first one is a  $n$ -bus radial network with height equal to one while the other is a five-bus network with a four-bus ring. The former gives ideas how many maximal cliques share a common bus. The latter demonstrates multiple-level bus and edge sharing. We will also show where the different components are implemented and how the communication is accomplished.

### A. n-Bus Radial Network

Consider the topology of the network given in Fig. 1(a). We have

$$\mathbf{M} = \begin{pmatrix} M_{11} & 0 & 0 & \cdots & M_{1n} \\ 0 & M_{22} & 0 & \cdots & M_{2n} \\ 0 & 0 & M_{33} & \cdots & M_{3n} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ M_{n1} & M_{n2} & M_{n3} & \cdots & M_{nn} \end{pmatrix}.$$

1) *Primal Algorithm:* Eq. (7) becomes

$$\text{Tr}(\mathbf{MW}) = \sum_{l=1}^{n-1} \underbrace{(M_{nl}^H W_{nl} + M_{ln}^H W_{ln} + M_{ll}^H W_{ll})}_{\text{unique terms}} + \underbrace{M_{nn}^H W_{nn}}_{\text{shared term}}$$

Each branch with the end nodes forms a maximal clique. We have  $\mathcal{C} = \{C_l | 1 \leq l \leq n-1\}$  where  $C_l = \{l, n\}$  and  $\mathcal{U} = (n, n)$ . By introducing a slack variable  $X_{nn,l}$  for  $W_{nn}$  to  $C_l$ , we have

$$\tilde{\mathbf{W}}_{C_l C_l} = \begin{pmatrix} W_{ll} & W_{ln} \\ W_{nl} & X_{nn,l} \end{pmatrix}.$$

Let

$$\mathbf{M}_l = \begin{pmatrix} M_{ll} & M_{ln} \\ M_{nl} & 0 \end{pmatrix}.$$

Given  $W_{nn}$ , subproblem  $l$  for  $C_l$  is stated as

$$\begin{aligned} & \text{minimize} && \text{Tr}(\mathbf{M}_l \tilde{\mathbf{W}}_{C_l C_l}) \\ & \text{subject to} && \underline{V}_l^2 \leq \text{Tr} \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix} \tilde{\mathbf{W}}_{C_l C_l} \leq \bar{V}_l^2 \\ & && \text{Tr} \begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix} \tilde{\mathbf{W}}_{C_l C_l} = W_{nn} \\ & && \tilde{\mathbf{W}}_{C_l C_l} \succeq 0 \end{aligned} \quad (34)$$

which is an SDP with a  $2 \times 2$  variable. Recall that  $\phi_l(W_{nn})$  is the optimal value of subproblem  $l$  given  $W_{nn}$ . The master problem is

$$\begin{aligned} & \text{minimize} && \sum_{l=1}^{n-1} \phi_l(W_{nn}) + M_{nn}^H W_{nn} \\ & \text{subject to} && \underline{V}_n^2 \leq W_{nn} \leq \bar{V}_n^2. \end{aligned}$$

$\lambda_{nn,l}$  is the Lagrangian multiplier for equality  $X_{nn,l} = W_{nn}$  of subproblem  $l$ . We update  $W_{nn}$  by

$$W_{nn}^{(t+1)} = \text{Proj} \left( W_{nn}^{(t)} - \alpha^{(t)} \left( \sum_{l=1}^{n-1} (-\lambda_{nn,l}) + M_{nn}^H \right) \right) \quad (35)$$

in  $[\underline{V}_n^2, \bar{V}_n^2]$ .

In iteration  $t$ , bus  $n$  broadcasts  $W_{nn}^{(t)}$  to bus  $l$ ,  $1 \leq l \leq n-1$ . After receiving  $W_{nn}^{(t)}$ , for all  $l$ , bus  $l$  solves its own subproblem (34) by any suitable SDP method, e.g. primal-dual IPM [10], in parallel and then returns  $\lambda_{nn,l}$  to bus  $n$ .<sup>2</sup> After receiving all  $\lambda_{nn,l}$ , node  $n$  updates  $W_{nn}$  by (35). The communication pattern is shown in Fig. 1(b).

<sup>2</sup>For most of the interior-point methods, primal and dual solutions come in pair. When the algorithm finds the optimal  $\tilde{\mathbf{W}}_{C_l C_l}$ , it will also give  $\lambda_{nn,l}$ . Thus no extra calculation is required to determine  $\lambda_{nn,l}$ .

2) *Dual Algorithm:* Eq. (19) becomes

$$\text{Tr}(MW) = \sum_{l=1}^{n-1} (M_{nl}^H W_{nl} + M_{ln}^H W_{ln} + M_{ll}^H W_{ll}) + \frac{M_{nn}^H W_{nn}}{n-1}$$

As only  $W_{nn}$  is common to all maximal cliques, we have  $\Omega_{nn} = \{C_1, \dots, C_{n-1}\}$  and

$$X_{nn,1} = X_{nn,2} = \dots = X_{nn,n-1} = W_{nn}. \quad (36)$$

Assume that (36) is arranged as follows. We assign Lagrangian multipliers to the equalities and we have

$$\begin{aligned} v_{nn,1}(X_{nn,1} - X_{nn,2}) &= 0 \\ v_{nn,2}(X_{nn,1} - X_{nn,3}) &= 0 \\ &\vdots \\ v_{nn,n-2}(X_{nn,1} - X_{nn,n-1}) &= 0. \end{aligned} \quad (37)$$

For  $1 \leq l \leq n-2$ ,  $\tilde{X}_{nn,l}(1) = X_{nn,1}$  and  $\tilde{X}_{nn,l}(2) = X_{nn,l+1}$ . Then we have

$$\begin{aligned} \tilde{v}_{nn,1} &= v_{nn,1} + \dots + v_{nn,n-2}, \\ \tilde{v}_{nn,l} &= -v_{nn,l-1}, \quad 2 \leq l \leq n-1. \end{aligned}$$

Let

$$\mathbf{M}_l = \begin{pmatrix} M_{ll} & M_{ln} \\ M_{nl} & \tilde{M}_l \end{pmatrix}$$

where

$$\tilde{M}_l = \begin{cases} \left( \frac{M_{nn}}{n-1} + v_{nn,1} + \dots + v_{nn,n-2} \right), & l = 1 \\ \left( \frac{M_{nn}}{n-1} - v_{nn,l-1} \right), & 2 \leq l \leq n-1 \end{cases}$$

and

$$\tilde{\mathbf{W}}_{C_l C_l} = \begin{pmatrix} W_{ll} & W_{ln} \\ W_{nl} & X_{nn,l} \end{pmatrix}.$$

Subproblem  $l$  is stated as

$$\begin{aligned} & \text{minimize} && \text{Tr}(\mathbf{M}_l \tilde{\mathbf{W}}_{C_l C_l}) \\ & \text{subject to} && \underline{V}_l^2 \leq \text{Tr} \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix} \tilde{\mathbf{W}}_{C_l C_l} \leq \bar{V}_l^2 \\ & && \underline{V}_n^2 \leq \text{Tr} \begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix} \tilde{\mathbf{W}}_{C_l C_l} \leq \bar{V}_n^2 \\ & && \tilde{\mathbf{W}}_{C_l C_l} \succeq 0 \end{aligned} \quad (38)$$

which is also an SDP with a  $2 \times 2$  variable. We update the price by, for  $1 \leq r \leq n-2$ ,

$$v_{nn,r}^{(t+1)} = v_{nn,r}^{(t)} - \alpha^{(t)} (X_{nn,r+1} - X_{nn,1}). \quad (39)$$

At time  $t$ , node  $l$  from  $C_l$ ,  $1 \leq l \leq n-1$ , solves (38)<sup>3</sup>, e.g., by IPM, in parallel and sends  $X_{nn_l}$  from the optimal  $\tilde{\mathbf{W}}_{C_l C_l}$  to bus  $n$ . Whenever any pair of  $X_{nn}$  specified in (47) (e.g.,  $X_{nn,1}$  and  $X_{nn,l}$  for  $C_1$  and  $C_l$ , respectively) reach bus  $n$ , price  $v_{nn,l-1}$  can be updated with (39) by bus  $n$  and the updated  $v_{nn,l-1}$  is multicast back to the corresponding buses (e.g., nodes 1 and  $l$ ). The communication pattern is shown in Fig. 1(b).

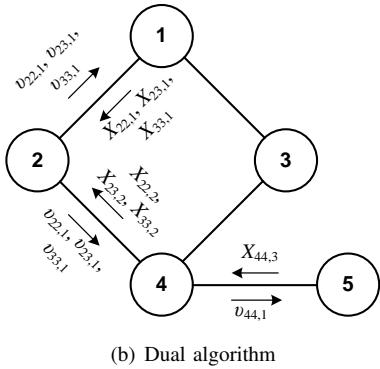
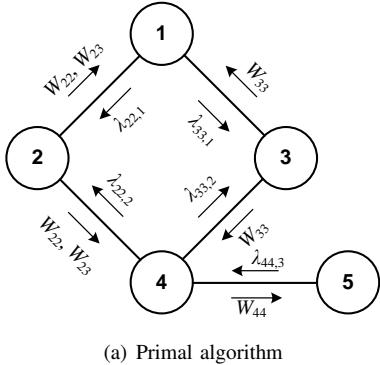


Fig. 2. Structure and communication patterns for the five-bus network with a four-bus ring

### B. Five-Bus Network with a Four-Bus Ring

Consider the topology of the network given in Fig. 2(a). We have

$$\mathbf{M} = \begin{pmatrix} M_{11} & M_{12} & M_{13} & 0 & 0 \\ M_{21} & M_{22} & 0 & M_{24} & 0 \\ M_{31} & 0 & M_{33} & M_{34} & 0 \\ 0 & M_{42} & M_{43} & M_{44} & M_{45} \\ 0 & 0 & 0 & M_{54} & M_{55} \end{pmatrix}$$

To make it chordal, suppose we add a fill-in edge between buses 2 and 3 and we have  $C_1 = \{1, 2, 3\}$ ,  $C_2 = \{2, 3, 4\}$ , and  $C_3 = \{4, 5\}$ . Assume bus 2 is used to co-ordinate  $C_1$  and  $C_2$ , and bus 4 for  $C_2$  and  $C_3$ .

1) *Primal Algorithm:* Let

$$\begin{aligned} \tilde{\mathbf{W}}_{C_1 C_1} &= \begin{pmatrix} W_{11} & W_{12} & W_{13} \\ W_{21} & X_{22,1} & X_{23,1} \\ W_{31} & X_{23,1} & X_{33,1} \end{pmatrix}, & \mathbf{M}_1 &= \begin{pmatrix} M_{11} & M_{12} & M_{13} \\ M_{21} & 0 & 0 \\ M_{31} & 0 & 0 \end{pmatrix}, \\ \tilde{\mathbf{W}}_{C_2 C_2} &= \begin{pmatrix} X_{22,2} & X_{23,2} & W_{24} \\ X_{32,2} & X_{33,2} & W_{34} \\ W_{42} & W_{43} & X_{44,2} \end{pmatrix}, & \mathbf{M}_2 &= \begin{pmatrix} 0 & 0 & M_{24} \\ 0 & 0 & M_{34} \\ M_{42} & M_{43} & 0 \end{pmatrix}, \\ \tilde{\mathbf{W}}_{C_3 C_3} &= \begin{pmatrix} X_{44,3} & W_{45} \\ W_{54} & W_{55} \end{pmatrix}, & \mathbf{M}_3 &= \begin{pmatrix} 0 & M_{45} \\ M_{54} & M_{55} \end{pmatrix}. \end{aligned}$$

We have

<sup>3</sup>In fact, any bus in a maximal clique can be elected to solve the subproblem. In this case, node  $l$  is chosen to reduce the computation concentrated at bus  $n$ .

- Subproblem 1: Given  $W_{22}$  and  $W_{33}$ ,

$$\begin{aligned} &\text{minimize} && \text{Tr}(\mathbf{M}_1 \tilde{\mathbf{W}}_{C_1 C_1}) \\ &\text{subject to} && \underline{V}_1^2 \leq W_{11} \leq \bar{V}_1^2 \\ &&& X_{ii,1} = W_{ii}, i = 2, 3 \\ &&& \tilde{\mathbf{W}}_{C_1 C_1} \succeq 0. \end{aligned} \quad (40)$$

- Subproblem 2: Given  $W_{22}$ ,  $W_{33}$ , and  $W_{44}$ ,

$$\begin{aligned} &\text{minimize} && \text{Tr}(\mathbf{M}_2 \tilde{\mathbf{W}}_{C_2 C_2}) \\ &\text{subject to} && X_{ii,2} = W_{ii}, i = 2, 3, 4 \\ &&& \tilde{\mathbf{W}}_{C_2 C_2} \succeq 0. \end{aligned} \quad (41)$$

- Subproblem 3: Given  $W_{44}$ ,

$$\begin{aligned} &\text{minimize} && \text{Tr}(\mathbf{M}_3 \tilde{\mathbf{W}}_{C_3 C_3}) \\ &\text{subject to} && \underline{V}_5^2 \leq W_{55} \leq \bar{V}_5^2 \\ &&& X_{44,3} = W_{44} \\ &&& \tilde{\mathbf{W}}_{C_3 C_3} \succeq 0. \end{aligned} \quad (42)$$

In iteration  $t$ , bus 2 sends  $W_{22}$  and  $W_{23}$  to both buses 1 and 4 and bus 3 sends  $W_{33}$  to both buses 1 and 4. Bus 4 sends  $W_{44}$  to bus 5. Buses 1, 4, and 5 compute (40), (41), and (42), respectively. Then bus 1 sends  $\lambda_{22,1}$  to bus 2 and  $\lambda_{33,1}$  to bus 3. Bus 4 sends  $\lambda_{22,2}$  to bus 2 and  $\lambda_{33,2}$  to bus 3. Bus 5 sends  $\lambda_{44,3}$  to bus 4, which has  $\lambda_{44,2}$ . Buses 2 and 3 update  $W_{ii}$  by

$$W_{ii}^{(t+1)} = \text{Proj} \left( W_{ii}^{(t)} - \alpha^{(t)} (-\lambda_{ii,1} - \lambda_{ii,2} + M_{ii}^H) \right) \quad (43)$$

within  $[\underline{V}_i^2, \bar{V}_i^2]$ , for  $i = 2, 3$ , respectively. Bus 4 updates  $W_{44}$  by

$$W_{44}^{(t+1)} = \text{Proj} \left( W_{44}^{(t)} - \alpha^{(t)} (-\lambda_{44,2} - \lambda_{44,3} + M_{44}^H) \right) \quad (44)$$

within  $[\underline{V}_4^2, \bar{V}_4^2]$ . Bus 2 updates  $W_{23}$  by

$$\text{Re}\{W_{23}^{(t+1)}\} = \text{Re}\{W_{23}^{(t)}\} - \alpha^{(t)} (-\lambda_{23,1}^{\text{Re}} - \lambda_{23,2}^{\text{Re}}), \quad (45)$$

and

$$\text{Im}\{W_{23}^{(t+1)}\} = \text{Im}\{W_{23}^{(t)}\} - \alpha^{(t)} (-\lambda_{23,1}^{\text{Im}} - \lambda_{23,2}^{\text{Im}}). \quad (46)$$

The communication pattern is shown in Fig. 2(a).

2) *Dual Algorithm:* Assume the equalities for the slack variables are arranged as follows. With Lagrangian multipliers, we have

$$\begin{aligned} v_{22,1}(X_{22,1} - X_{22,2}) &= 0 \\ v_{23,1}(X_{23,1} - X_{23,2}) &= 0 \\ v_{33,1}(X_{33,1} - X_{33,2}) &= 0 \\ v_{44,1}(X_{44,2} - X_{44,3}) &= 0. \end{aligned} \quad (47)$$

$$\mathbf{M}_1 = \begin{pmatrix} M_{11} & M_{12} & M_{13} \\ M_{21} & \frac{M_{22}}{2} + v_{22,1} & \frac{M_{23}}{2} + v_{23,1} \\ M_{31} & \frac{M_{32}}{2} + v_{23,1} & \frac{M_{33}}{2} + v_{33,1} \end{pmatrix},$$

$$\mathbf{M}_2 = \begin{pmatrix} \frac{M_{22}}{2} - v_{22,1} & \frac{M_{23}}{2} - v_{23,1} & M_{24} \\ \frac{M_{32}}{2} - v_{23,1} & \frac{M_{33}}{2} - v_{33,1} & M_{34} \\ M_{42} & M_{43} & \frac{M_{44}}{2} + v_{44,1} \end{pmatrix},$$

TABLE I  
NORMALIZED CPU TIME FOR DISTRIBUTION TEST FEEDERS<sup>a</sup>

Number of buses	Centralized	Cumulative		Distributed	
		Primal	Dual	Primal	Dual
8	1.85	7.21	5.62	1.52	1.00
34	298.68	37.79	33.89	1.94	1.70
123	— <sup>b</sup>	143.39	126.48	2.24	1.64

<sup>a</sup> The CPU times are normalized by 0.0857s.

<sup>b</sup> The solver cannot be applied because of the out-of-memory problem.

and

$$\mathbf{M}_3 = \begin{pmatrix} \frac{M_{44}}{2} - v_{44,1} & M_{45} \\ M_{54} & M_{55} \end{pmatrix}.$$

We have

- Subproblem 1: Given  $v_{22,1}$ ,  $v_{23,1}$ , and  $v_{33,1}$ ,

$$\begin{aligned} & \text{minimize} && \text{Tr}(\mathbf{M}_1 \tilde{\mathbf{W}}_{C_1 C_1}) \\ & \text{subject to} && \underline{V}_i^2 \leq W_{ii} \leq \bar{V}_i^2, i = 1, 2, 3 \\ & && \tilde{\mathbf{W}}_{C_1 C_1} \succeq 0. \end{aligned} \quad (48)$$

- Subproblem 2: Given  $v_{22,1}$ ,  $v_{23,1}$ ,  $v_{33,1}$ , and  $v_{44,1}$ ,

$$\begin{aligned} & \text{minimize} && \text{Tr}(\mathbf{M}_2 \tilde{\mathbf{W}}_{C_2 C_2}) \\ & \text{subject to} && \underline{V}_i^2 \leq W_{ii} \leq \bar{V}_i^2, i = 2, 3, 4 \\ & && \tilde{\mathbf{W}}_{C_2 C_2} \succeq 0. \end{aligned} \quad (49)$$

- Subproblem 3: Given  $v_{44,1}$ ,

$$\begin{aligned} & \text{minimize} && \text{Tr}(\mathbf{M}_3 \tilde{\mathbf{W}}_{C_3 C_3}) \\ & \text{subject to} && \underline{V}_i^2 \leq W_{ii} \leq \bar{V}_i^2, i = 4, 5 \\ & && \tilde{\mathbf{W}}_{C_3 C_3} \succeq 0. \end{aligned} \quad (50)$$

At time  $t$ , bus 2 announces  $v_{22,1}$ ,  $v_{23,1}$ , and  $v_{33,1}$  to buses 1 and 4, and bus 4 sends  $v_{44,1}$  to bus 5. Buses 1, 4, and 5 solve (48), (49), and (50), respectively. Then bus 1 sends  $X_{22,1}$ ,  $X_{23,1}$ , and  $X_{33,1}$  to bus 2. Bus 4 sends  $X_{22,2}$ ,  $X_{23,2}$ , and  $X_{33,2}$  to bus 2. Bus 5 sends  $X_{44,3}$  to bus 4. Bus 2 updates  $v_{22,1}$ ,  $v_{23,1}$  and  $v_{33,1}$  by

$$v_{ik,1}^{(t+1)} = v_{ik,1}^{(t)} - \alpha^{(t)} (X_{ik,2} - X_{ik,1}), i, k = 2, 3, i \leq k, \quad (51)$$

and bus 4 updates  $v_{44,1}$  by

$$v_{44,1}^{(t+1)} = v_{44,1}^{(t)} - \alpha^{(t)} (X_{44,3} - X_{44,2}). \quad (52)$$

The communication pattern is shown in Fig. 2(b).

## VI. SIMULATION RESULTS

To evaluate the performance of the algorithms, we perform extensive simulations on various network settings. Since OPF is formulated as an SDP, the optimal solution can be computed in polynomial time by any popular SDP algorithms, e.g. IPM. Recall that the primal or dual algorithm aims to divide the original problem into smaller ones and to coordinate the subproblems, which of each can be solved by any SDP solver independently. The primal and dual algorithms perform coordination by indicating what problem data should be allocated to each subproblem and do simple calculations to update the shared terms (for the primal) and the prices (for the dual).

TABLE II  
NORMALIZED CPU TIME FOR IEEE POWER TRANSMISSION SYSTEM BENCHMARKS<sup>a</sup>

Number of buses	Centralized	Cumulative dual	Distributed dual	iterations	Initial step size
14	5.38	5.38	1.00	1	30
30	45.29	58.60	5.38	6	30
57	1696.79	49.08	4.28	4	30
118	— <sup>b</sup>	704.46	13.51	9	300

<sup>a</sup> The simulations for this problem set are done on MacBook Pro with 2.4 GHz Intel core i5 and 4 GB RAM. The CPU times are normalized by 0.1410s.

<sup>b</sup> The solver cannot be applied because of the out-of-memory problem.

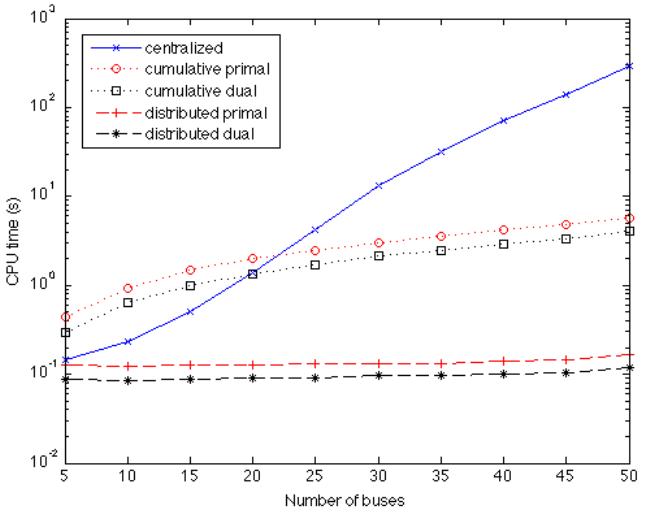


Fig. 3. CPU time of the various approaches on radial networks with bounded voltages

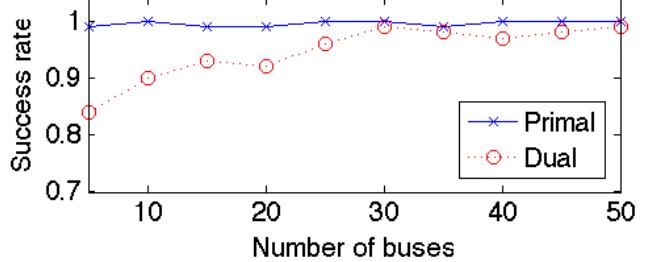


Fig. 4. Success rates of the primal and dual algorithms on radial networks with bounded voltages

When compared with those done by the SDP solver, the computation and ordination required solely by our algorithms are relatively far less stringent. As a whole, the bottleneck of computation should be at the SDP solver. In our simulation, we program the primal and dual algorithms in MATLAB and solve each SDP with YALMIP [20] and SeDuMi [21]. To get rid of the dependence on the programming language and to simplify the comparison, we only count the CPU time spent on the SDP solver. Moreover, we can arrange the subproblems to be solved in a single node or distribute

them to different nodes in the network. For the former, we assume the problems are handled sequentially and we call it the cumulative approach. The latter, named as the distributed approach, addresses the subproblems in parallel. Without our algorithms, the (original) problem will be solved in its original form in a centralized manner. Here we compare the CPU times required for the SDP(s) among the centralized approach, (primal and dual) cumulative approaches, and (primal and dual) distributed approaches.

We run the simulations on Dell PowerEdge 2650 with  $2 \times 3.06\text{GHz}$  Xeon and 6GB RAM (except those for the transmission system benchmarks in Table II).<sup>4</sup> In order to monitor the performance in each simulation run, we assemble the partial solutions (done by the subproblems) to form a complete one for the original problem and evaluate the corresponding objective value.<sup>5</sup> Our algorithms stop when the computed objective value falls in the range of  $10^{-2} \times$  the global minimum. We assume that the dual algorithm is synchronized. In other words, all subproblems for the dual are solved in each iteration (but this is not required when implemented in real systems). The initial step size  $\alpha^{(0)}$  is set to one and it is updated by  $\alpha^{(t)} = \alpha^{(t-1)}/t, \forall t > 0$ . An algorithm is deemed successful if the stopping criterion is met in 100 iterations.

We perform simulations on three problem sets; the first two focuses on tree-like networks while the last one is about transmission networks. The first problem set is some distribution test feeder benchmarks [22]. As the data set does not specify the cost function of power production/consumption, we create a problem instance by randomly generating the costs. To do this, we first select one node, e.g. node  $i$ , to be the power source node with  $c_{i1}$  set randomly in the range  $(0, 10)$ . For other node  $k \neq i$ ,  $c_{k1}$  set randomly in the range  $(-10, 0)$ . We create 100 instances for each network. Table I shows the averages of the normalized CPU times of the various approaches. All algorithms converge in 100 iterations for all instances.

The second problem set is the  $n$ -bus radial network demonstrated in Section V. For each instance, the root is the power source with a random cost selected in  $(0, 10)$  and each of the rest takes a random cost in  $(-10, 0)$ . For each node  $i$ , we specify a number  $\xi$  in  $(0.9, 1.1)$  and set  $\underline{V}_i = 0.95\xi$  and  $\bar{V}_i = 1.05\xi$ . For each line, the magnitudes of the conductance and susceptance are randomly assigned in  $(0, 10)$ . We produce 100 instances for each  $n$  and plot the average CPU times in logarithmic scale in Fig. 3.

From the simulation results for these two problem sets, both the primal and dual algorithms converge very fast for distribution networks. The CPU time for the centralized approach grows very fast with the size of the network. The CPU time grows roughly linearly for the cumulative approach while it becomes independent of the size for the distributed approach. We define success rate as the fraction of the total number of simulation runs with stopping criterion met in 100 iterations, shown in Fig. 4. The success rate of the primal is almost

<sup>4</sup>The results in Tables I and II are normalized, and thus, they are comparable.

<sup>5</sup>The assembly of partial solutions is not required in real implementation.

100% for all tested network sizes. The dual fails to converge in 100 iterations for a small fraction of small networks but the success rate grows to almost 100% with the network size. In general, the primal and dual algorithms are similar in performance but the dual requires a little bit less CPU time than the primal on the average.

The third problem set is some IEEE power transmission system benchmarks [23]. As pointed out in [3], these test cases have zero duality gap although they have network structures different from what we mention in Section II-B. Table II shows the CPU times required, the iterations for convergence, and the initial step sizes. The primal algorithm is not applied to this problem set and the reason will be given in the next section. For these transmission network topologies, the maximal cliques of the fill-in graphs are much irregular than those with tree-structured networks. There are many ways to construct the maximal cliques and different construction can result in different convergence speed. The study of the relationship between maximal clique construction and the algorithm performance is out of the scope of this paper. In this simulation, we randomly choose one maximal clique configuration and the step sizes are adjusted individually so as to have fast convergence. Nevertheless, the dual algorithm is more desirable than the centralized approach.

## VII. DISCUSSION

Both the primal and dual algorithms try to tackle the original problem by solving smaller subproblems but the ways to handle the information corresponding to the common partial solutions between subproblems are different. For the primal algorithm, if a common solution corresponds to a bus, that bus will compute its partial solution with the required information. For example, in Fig. 1(a),  $W_{nn}$  for bus  $n$  is common to all the subproblems. Bus  $n$  computes  $W_{nn}$  with its own  $\underline{V}_n$ ,  $\bar{V}_n$ , and  $M_{nn}$ . Only the computed  $W_{nn}$  is required to transmit to other buses which do not require bus  $n$ 's information. For the dual algorithm, each subproblem needs to acquire all its bus information, even for the common bus. Consider the example in Fig. 1(a) again, for  $1 \leq i \leq n-1$ , node  $i$  requires  $\underline{V}_n$ ,  $\bar{V}_n$ , and  $M_{nn}$  to solve its subproblem. Similarly, if the common solution is for a link, we can just elect any one of the connected bus to do the computation with the primal and dual algorithms. Therefore, the primal algorithm requires less information sharing between buses and it favors situations with sensitive bus information.

Our algorithms do not require an overlay of communication networks with different topology. From the examples in Section V, all the communication is one-hop. A node needs to transmit data, e.g.  $W, \lambda, X$ , and  $v$ , to its neighbor nodes only. Communication links need to be built along with existing transmission lines only.

The primal algorithm is suitable for networks with tree structure while the dual can handle those with rings. In fact, the primal is not very efficient to update the partial solutions for (fill-in) edges, i.e. (15)–(18), especially when their values are closed to the boundary of the feasible region. When updating the variables, the bus variables, i.e.  $W_{ii}$ , are bounded but it is

not the case for the edge variables, i.e.  $W_{ik}, i \neq k$ . When the step size is too large, (15)–(18) may drive the current point out of the feasible region. If so, the obtained  $\lambda$  cannot be used to form a subgradient, and thus (15)–(18) fail. When reaching an infeasible point from a feasible one, we can always step back and choose a smaller step for an update again. When approaching the optimal solution which is located on the boundary for SDP, this step-back procedure is not very efficient. The dual algorithm does not have this problem when updating  $v$  with (28). As mentioned, we can always constrain a feasible solution by averaging the shared variables.

Tree networks have a very nice property with respect to maximal clique construction; each branch with its attached nodes form a maximal clique. Hence, it is straightforward to decompose the problem into subproblems. However, when it comes to a more irregular network like those IEEE transmission system benchmarks, the number of ways to decompose the problem grows with the size of the network. The performance of our algorithms also depends on how we form the maximal cliques and the initial step size needs to be adjusted accordingly. For problems with tree structure, the performance is easier to predict and the algorithms converge faster.

The dual algorithm is more resistant to communication delay than the primal. For the primal, an update of a shared variable requires  $\lambda$  from all involved subproblems and thus synchronization is required. Delay of computing or transmitting an  $\lambda$  from any subproblem can affect the whole algorithm proceed. On the other hand, an update of an  $v$  requires the  $\tilde{X}$  from two pre-associated subproblems according to the arrangement of the inequalities in 22. Delay of computing or transmitting a particular  $\tilde{X}$  can affect the update of some but not all the  $v$ . Thus the dual algorithm is asynchronous. Moreover, we can pair the variables in (22) into equalities differently and secretly whenever we start the dual algorithm. In some sense, the dual algorithm is more robust to attack stemmed from communication on the communication links.

### VIII. CONCLUSION

OPF is very important in planning the schedule of power generation. In the smart grid paradigm, more renewable energy sources will be incorporated into the system, especially in distribution networks, and the problem size will also grow tremendously. As problems with some special structures (e.g. trees for distribution networks) have a zero duality gap, we can find the optimal solution by solving the convex dual problem. In this paper, we propose the primal and dual algorithms (with respect to the primal and dual decomposition techniques) to speed up the computation of the convexified OPF problem. The problem is decomposed into smaller subproblems, each of which can be solved independently and effectively. The primal algorithm coordinates the subproblems by controlling the shared terms (related to electricity resources) while the dual one manages them by updating the prices. From the simulation results for tree-structure problems, the computation time grows linearly with the problem size if we solve the decomposed problem in a central node with our algorithms. The computation time becomes independent of the problem

size when the subproblems are solved in parallel in different nodes. Even without nice network structure such as a tree, the dual algorithm outperforms the centralized approach without decomposition. Therefore, the primal and dual algorithms are excellent in addressing OPF, especially for distribution networks. In future, we will improve the algorithm by incorporate more constraints into the OPF problem and move to nonlinear objective functions.

### REFERENCES

- [1] P. P. Varaiya, F. F. Wu, and J. W. Bialek, “Smart operation of smart grid: Risk-limiting dispatch,” *Proc. IEEE*, vol. 99, pp. 40–57, 2011.
- [2] B. Stott, J. Jardim, and O. Alsac, “Dc power flow revisited,” *IEEE Trans. Power Syst.*, vol. 24, pp. 1290–1300, 2009.
- [3] J. Lavaei and S. H. Low, “Zero duality gap in optimal power flow problem,” *IEEE Trans. Power Syst.*, in press.
- [4] B. Zhang and D. Tse, “Geometry of feasible injection region of power networks,” *To appear in proc. Allerton*, 2011.
- [5] S. Sojoudi and J. Lavaei, “Network topologies guaranteeing zero duality gap for optimal power flow problem,” *Submitted to IEEE Trans. Power Syst.*, 2011.
- [6] R. A. Jabr, “Radial distribution load flow using conic programming,” *IEEE Trans. Power Syst.*, vol. 21, pp. 1458–1459, 2006.
- [7] F. Kelly, A. Maulloo, and D. Tan, “Rate control in communication networks: shadow prices, proportional fairness and stability,” *Journal of the Operational Research Society*, vol. 49, pp. 237–252, 1998.
- [8] D. P. Bertsekas, *Nonlinear programming*, 2nd ed. Belmont, MA, USA: Athena Scientific, 1999.
- [9] M. Chiang, S. H. Low, A. R. Calderbank, and J. C. Doyle, “Layering as optimization decomposition: A mathematical theory of network architectures,” *Proc. IEEE*, vol. 95, pp. 255–312, Jan. 2007.
- [10] S. Wright, *Primal-dual interior-point methods*. Philadelphia, PA, USA: Society for Industrial and Applied Mathematics, 1997.
- [11] R. E. Tarjan and M. Yannakakis, *Simple linear-time algorithms to test chordality of graphs, test acyclicity of hypergraphs, and selectively reduce acyclic hypergraphs*. Philadelphia, PA, USA: Society for Industrial and Applied Mathematics, July 1984, vol. 13.
- [12] D. R. Fulkerson and O. A. Gross, “Incidence matrices and interval graph,” *Pacific J. Math.*, vol. 15, no. 3, pp. 835–855, 1965.
- [13] R. E. Neapolitan, *Probabilistic reasoning in expert systems: theory and algorithms*. New York, NY, USA: John Wiley & Sons, Inc., 1990.
- [14] E. A. Akkoyunlu, “The enumeration of maximal cliques of large graphs,” *SIAM Journal on Computing*, vol. 2, pp. 1–6, 1973.
- [15] K. Nakata, M. Yamashita, K. Fujisawa, and M. Kojima, “A parallel primal-dual interior-point method for semidefinite programs using positive definite matrix completion,” *Parallel Comput.*, vol. 32, pp. 24–43, Jan. 2006.
- [16] M. Fukuda, M. Kojima, K. Murota, and K. Nakata, “Exploiting sparsity in semidefinite programming via matrix completion I: General framework,” *SIAM J. on Optimization*, vol. 11, pp. 647–674, March 2000.
- [17] K. Nakata, K. Fujisawa, M. Fukuda, M. Kojima, and K. Murota, “Exploiting sparsity in semidefinite programming via matrix completion II: implementation and numerical results,” *Mathematical Programming*, vol. 95, pp. 303–327, 2003.
- [18] R. Grone, C. R. Johnson, E. M. Sa, and H. Wolkowicz, “Positive definite completions of partial hermitian matrices,” *Linear Algebra and Its Applications*, vol. 58, pp. 109–124, 1984.
- [19] S. Boyd and L. Vandenberghe, *Convex Optimization*. New York, NY, USA: Cambridge University Press, 2004.
- [20] J. Löfberg, “YALMIP : A toolbox for modeling and optimization in MATLAB,” in *Proc. International Symposium on Computer Aided Control Systems Design*, Sep. 2004, pp. 284–289.
- [21] J. F. Sturm, “Using sedumi 1.02, a matlab toolbox for optimization over symmetric cones,” 1998.
- [22] W. H. Kersting, “Radial distribution test feeders,” in *Proc. IEEE Power Engineering Society Winter Meeting*, vol. 2, 2001, pp. 908–912.
- [23] University of washington, power systems test case archive. [Online]. Available: <http://www.ee.washington.edu/research/pstca>